Assembler Report

The purpose of this report is to describe the approach, problems and solutions of the Assembler Program for assignment 3. The task was to create an assembly program which converts assembly language into machine code. I used C++ to build the source code for this program.

To understand how to start building the program, I had to delve further into the fetch execute cycle. I took a test-driven approach by coding each function separately and testing to see if it functioned correctly and if so I would implement it into the final solution and test to ensure that the functions worked together properly.

One of the first problems I encountered was the coding language itself, I do not have as much experience in C++ than other languages such as C# and the syntax is much different. I uses lab time and worked at home to try and learn C++ as best as I could before I had to start coding.

During coding itself, the main problem I encountered was converting the assembly code into machine code, I was trying to convert the full file into machine code without removing comments and blank lines, I then decided it would be best to remove comments and such to just leave the code itself, this allowed for easier reading of the code and made to conversion much easier.

Another problem encountered was with a header file which contained the programs functions and variables, however this seemed to break the code as the main function was not working properly and would not call the appropriate functions or variables. I decided it would be best if I was to just have the full code in the one place which sorted the problem.

A problem with the instruction set caused the program to not work correctly. Originally I had only two columns in instruction set array to hold the opcode and the binary value, however when it came to converting the code it seemed that I needed a third column containing a Yes/No option which allowed for the use of Booleans in the program solving this problem.

To run the assembler program, the appropriate syntax is g++ -std=c++11 -o Assembler Assembler.cpp. This will compile and output the source code into this file. From there type ./Assembler to run the output and the code will run.